(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 January 2002 (10.01.2002)

PCT

(10) International Publication Number
WO 02/03645 A2

(51) International Patent Classification[7]: H04L 29/00

(21) International Application Number: PCT/US01/20411

(22) International Filing Date: 27 June 2001 (27.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/607,718 30 June 2000 (30.06.2000) US

(71) Applicant: NOKIA INC. [US/US]; 6000 Connection Drive, MS 1-5-744, Irving, TX 75039 (US).

(72) Inventor: MARKS, Bennett; 439 Bedford Road, Carlisle, MA 01741 (US).

(74) Agents: ROLNIK, Robert, C. et al.; Nokia Inc., 6000 Connection Drive, MS 1-4-755, Irving, TX 75119 (US).

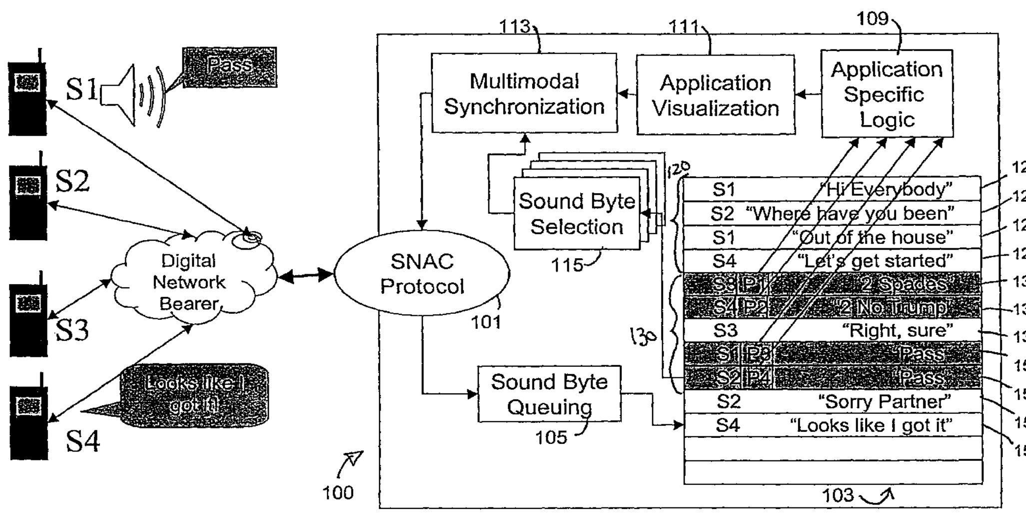
(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS TO SYNCHRONIZE AUDIO AND VISUAL APPLICATION DATA PRESENTATION



(57) Abstract: A server (100) is disclosed which may receive communications having both audio data and application data. Application data may illustrate progress of conversations or games. The server is able to filter such data according to an application (109), and transmit each communication to participating clients that have joined in the conversation or game, except not to the originator of the communication.

WO 02/03645 A2

METHOD AND APPARATUS TO SYNCHRONIZE AUDIO AND VISUAL APPLICATION DATA PRESENTATION

Field of the Invention

This invention relates non-real-time transmittal of multimedia data with application data, and more particularly to a method of transmitting voice data with application data and simultaneous presentation of the voice and application data on a client device.

Background of the Invention

For many years, the need to exchange information amongst a group has required the physical presence of all group members in a room so that genuine consensus, camaraderie, and contributions could be made in relation to the common goals of the group. The invention of the telephone has provided great opportunities for people not in the room to contribute -- and even more so with the availability of speaker-phone devices.

Common use radio channels have provided an alternate medium by which multiple people could speak and contribute, although not at a common location. Typically, e.g. in a Citizens Band radio environment, the one who transmitted on a channel with the strongest signal, and/or nearest location, would drown out any other voices contending for the channel.

Both of the foregoing communications means still only had audio by which to convey feelings, and changes in the environments within earshot of the communicating device. The continuing development in the 1980's and 1990's of the internet, particularly the use of WWW browsers with audio plug-ins, gave great new dimensions to the level of interactivity, and non-audio exchanges that are possible. With the availability of large bandwidths typical in wired, high-processing-capacity clients, volumes of digitally encoded speech and real-time graphics are now regularly exchanged between participants on the net.

IP telephony has provided an amenable substitute for analog phone conversations, mainly due to the real-time transmittal of full-duplex speech using protocols such as H.232. Unfortunately, such services require that the intermediate routers provide assured levels of delay, which comes at a cost of relegating other traffic to queues in routers and increasing the attendant delays for competing application traffic. Consequently, the costs of establishing and maintaining traffic for such real-time communications using packet networks is high for a one-to-one interface of people compared to the typical pattern of packet data usage. There is

corresponding geometric progression in traffic use for each increment of one person who participates in the voice chat or IP telephone conference.

Some shared mediums are particularly costly to maintain, e.g. the wireless bands supporting mobile phone and wireless data use. These bands must be allocated to functions associated with paging, call-setup and tear-down, roaming, handovers and power control, among others. The prospect of transmitting video over such mediums will make the radio bands more highly contended for, and increase the cost and value of this vital asset to wireless carriers. On the other hand, video and voice traffic by this medium is often very bursty, which leaves a valuable resource partially used for intermittent and largely random intervals. Consequently, there is available bandwidth for applications or services that can tolerate random delays while waiting for gaps in use by high priority traffic. Some such applications tolerant to delays are social game playing, and question and answer sessions between e.g. lecturers, and e.g. students. In each of the foregoing applications, delays of up to 10 seconds can be tolerated between, either, a) a movement of a game-piece, e.g. a card; or b) uttering a sound byte. Moreover, these applications can frequently be accompanied by periods of silence while players contemplate strategies, or students take notes. Consequently the full duplex of IP telephony is unnecessary.

One perceived drawback in conventional voice chat conducted to support playing of games, is that the application data is so compact in relation to any associated voice inputs, that the application data can arrive at a client and be represented graphically, before significant audio data is obtained. A triumphant cheer by a player upon achieving a small victory on the game surface or environment has a markedly diminished impact if the cheer occurs more than a few seconds after changes in the graphics reveal the move to the other players of the game.

Summary of the Invention

An embodiment of the invention provides a server-based mechanism for transmitting audio information paired with contextual application data, e.g. game status, to one or client devices that have subscribed or joined to the conversation hosted on a server. In many cases the client devices will be operated by persons or users. The paired audio and contextual information, also known as Serialized Network Asynchronous Communication (SNAC), may be provided in the sequence of the progress of the conversation, or by random access. The server stores each SNAC, and filters or modifies it according to any rules of the application, which may be a game program. When a user requests the information in a SNAC, the server transmits the SNAC.

An embodiment of the invention is a server device that mediates delivery of one or more Serialized Network Asynchronous Communications (SNAC). The server device receives a SNAC from a client device. The server stores the SNAC in storage. The server filters the SNAC according to the rules of an application to produce a second SNAC having application data and audio data. The server transmits the second SNAC via a packet switched network to at least one requesting client device. The server is responsive to 'request' commands made by client devices that have previously successfully ordered a 'join' command to the server. With each request, the server may provide a SNAC that has been modified in accordance with an application. Applications may be any kind of group interaction that depends on audio for part of its entertainment or educational value. A request may be for a SNAC that is expected but not yet stored at the server. A server typically does not send a SNAC as a response to a 'request' if the SNAC was also authored via the client device that is making the 'request'.

A second embodiment of the invention is a wireless client device that presents audio data of a SNAC and application data of a SNAC. The client device wirelessly receives a SNAC transported by at least two packets. The client renders the audio data. The client renders the application data, which could represent cards in game play, at about the same time the audio data is presented, delaying rendering of application data if necessary.

The client device may perform filtering of the audio data to alter its duration, among other things. The client may be built using a suitable environment or operating system, such as, for example, the Wireless Application Environment (WAE). The client may depend upon drivers or a Wireless Application Protocol (WAP) stack to operate as an interface to a wireless transceiver for purposes of packet assembly and disassembly, as well as selection of at least one bearer service.

Devices such as speakers and microphones may provide means to record and playback audio information of a SNAC in conjunction with a COder and DECoder (CODEC). A display, which may be a flat panel, ray-tube, or solid-state display, may provide a output for the display of aspects of the application data.

Among the many advantages of the present invention, one or more of the disclosed embodiments provides a way to couple the manipulation of application data so that the application data is presented at the same time that audio data is rendered. The application data may be data elements representing game-pieces, or conversation contexts. A way to store audio data that may arrive considerably delayed and at a rate slower than real-time so that low-speed bearer service may be

used is disclosed.

Another advantage provided by one or more embodiments includes the ability to maintain time based ordered relationships between the various parties involved in a group interaction. This may be accomplished by sorting at a server various SNACs that apply to a conversation based on a time-stamp of when the SNAC was sent or received.

Another advantage provided by one or more embodiments is that by coupling audio tightly with application data presentation, a quick, yet not real-time way of authoring, transmitting and presenting human inputs to a group is accomplished without the intensive use of bearer resources that might accompany a duplex audio conversation. That is, an exchange amongst identically sized groups may be accomplished more economically using an embodiment of the invention, than through real-time duplexed communications, and not suffer any confusion for certain applications, such as game playing. Although the competing cellular telephone audio data sent using a vox-limited CODEC, is of comparable size -- unlike typical digital cellular coding schemes of today, the audio of the embodiments may be sent piecemeal, as packets, without the high-speed requirements of low latency and low jitter associated with cellular telephony.

Moreover, any digital signal processor (DSP) that is running a CODEC on the client embodiment may operate in a less than real-time speed. It is permissible to have the network be a bottleneck for data flow at some times, and the DSP to be a bottleneck at other times. Reduced speeds and data throughput can operate to conserve battery life in a embodiment of the invention, which is particularly helpful if the client device embodiment has, as a primary function, full duplex, voice telephony functions too. In other words, the embodiment of the invention may be embedded in a device that performs voice telephony in a wireless manner.

While an aggregate of mobile phone users may increase data traffic using an embodiment of the invention, the use of celullar telephony spectrum to support traffic generated by the embodiments is likely to be compatible to being shared with duplexed voice telephony traffic on the same spectrum. This is because voice traffic of a user that owns a mobile phone is likely to be diminished during the duration of any game or conversation taking place using an embodiment.This invention makes appropriate use of scarce resources, such as wireless bandwidth, to deliver multi-modal application data, which might otherwise be deemed cost ineffective. The invention is in the area of optimized, cost-tuned, multimedia delivery in bandwidth limited environments and makes good use of a packet transmission medium that is subject to rapid variations in delivery speed.

Brief Description of the Drawings

The disclosed inventions will be described with reference to the accompanying drawings, which show important sample embodiments of the invention, wherein:

Fig. 1 shows a server embodiment of the invention and how the embodiment might interact with client devices;

Fig. 2 shows a Serialized Network Asynchronous Communication (SNAC) record used to store information in an embodiment;

Fig. 3 is a diagram of communications in an example of how a game application may function within a server embodiment of the invention;

Fig. 4 is a diagram of communications in an example of progress of a conversation or game;

Fig. 5 is a block diagram of a client device embodiment; and

Fig. 6 shows a state machine implementation of the SNAC protocol operating on a client device embodiment.

Detailed Description of the Preferred Embodiments

The numerous innovative teachings of the present application will be described with particular reference to the presently preferred embodiment. However, it should be understood that this class of embodiments provides only a few examples of the many advantageous uses of the innovative teachings herein. In general, statements made in the specification of the present application do not necessarily delimit any of the various claimed inventions. Moreover, some statements may apply to some inventive features but not to others.

A central aspect of the embodiments of the invention, is the ability to store, transmit, receive and present the data stored in a Serialized Network Asynchronous Communication (SNAC). The SNAC may carry both application data and audio, wherein one or more are compressed. A SNAC may be stored on a device as a record, which may be on non-volatile or volatile memory. In many instances, the audio data will be the predominate data of the SNAC record. Transmission of a SNAC record may be according to a packet switched protocol. Use of such a protocol may be slower than use of a circuit switched protocol. Moreover, it is anticipated that in some cases, an application data part of a SNAC may be received by a device well in advance of an audio portion.

A device that is memory frugal, may transmit a SNAC, and reallocate memory in steps as the transmission facilities of the network indicate to the device that proper transmittal of portions of the SNAC. In such an instance, the SNAC may exist in part a) in memory of the transmitting device; b) as packets transmitted by a packet

switching network; and c) in memory of the receiving device. A group of SNACs may be organized at a server into a conversation. A conversation is a set of SNACs that exist in furtherance of an application, such as a game, or a discussion.

Fig. 1 shows an embodiment of the invention and how the embodiment might interact with wireless clients, S1 **191**, S2 **192**, S3 **193** and S4 **194**. One of the chief advantages of using wireless clients, is the fact that wireless clients tend to be always on, and also tend to be nearer at hand more frequently, than, for example, a desk-bound computer. Consequently, a greater population of ready, willing and able game players will likely be available through such devices.

A server **100** device, including a CPU, provides facilities for program operation in support of client devices. Fig. 1 illustrates such a server. Serialized Network Asynchronous Communication (SNAC) protocol **101** operates as a data port to a packet network. The packet network may provide connectivity to a wireless data server, such as a Wireless Application Protocol (WAP) server, for reception and transmission of SNAC packets and packet series. The WAP server may be addressable over a wide area network of, e.g. a cellular telephone system. The WAP server may be a resource addressable via the Internet. WAP server may forward the SNAC to a wireless transceiver. Wireless transceiver transmits SNAC over a wireless carrier to a wireless client device. The SNAC protocol **101** handles matters such as determining to which conversation an incoming SNAC belongs, and providing opportunities to create new conversations. SNAC protocol **101** provides for limitations in the number of clients that enter a conversation, e.g. to limit the clients participating in a conversation to those allowed by the application specific logic **109**. SNAC Protocol **101** also provides a data port for receiving SNACs over a packet network, as well as for sending SNACs over a packet network.

SNAC protocol **101** places a received SNAC into a queue or other data structure or queue **103**, by use of a pointer **105** to the next open space in memory. Such memory may be in the form of RAM, disk storage, magnetic storage and need not be located on-site with the SNAC protocol **101**. The queue may hold information for as long as needed, and once the information has been disseminated to all members of the conversation/game, the memory allocated to it may be freed. Alternatively, data could simply reach a certain age after its creation, and that may trigger reallocation of the memory.

In the event that a SNAC contains no sound data, such an instance may be represented as null pointers, or other symbol in the data structure or queue**103**. Any time application data, such as P1 **107**, is put into the queue **103**, the application specific logic **109**, takes steps consistent with a application or program to orchestrate

rules of the game, whether that game be a card game, board game, trivia game or the like.

Application specific logic (ASL) **109** may make use of a data structure to track game-piece inventory, placement, points accrued and properties owned, among other things. ASL may be one of several processes operating on a CPU of the server. Changes may be such that a visual indication should be sent to one or more of the players to indicate game or conversation progress. For this purpose, application visualization **111** may provide the needed formatting of the changed conditions for each of the affected players. Such formatting may be in the form of XML, HTML, WML or raw text such that output of application visualization **111** is formatted data. Application visualization **111** presents formatted data to multi-modal synchronization **113.** Multi-modal synchronization**113** checks the data structure or queue **103** for any sound data associated with the formatted data. Frequently, the sound data associated with the formatted data will be in the last SNAC received, however, a player, who operates a client, may have designated a default sound data to be associated with one or more types of moves. In the latter case, a sound data could be repetitively selected from every instance where a player plays a card on a game surface in a card game application. A SNAC may have no sound data stored in the data structure or queue **103**, and so no voice data is retrieved by sound byte selection **115**. Sound byte selection **115** may be a collection of pointers, one for each client that has joined the conversation or game. The queue may be arranged so that SNACs are in order for oldest to youngest. A 'next' position in the queue **115** is a position that is younger and more recent in time than the current position. The fact that there is no sound data associated with an application data as modified by ASL is transmitted to multi-modal synchronization **113.** A player, who is making a request for a SNAC, may have selected not to receive the sound portion. If that is the case, the multi-modal synchronization may be set to ignore the sound byte selection **115** and transmit only formatted data. A number of rules may be set up controlling whether sound data should be ignored or modified. Such rules operating on the multi-modal synchronization **113** effectively filter a SNAC stored at the server **100.**

One example of an application that could be supported by the embodiment is an application of a four player card game. Each player operates a client device, which may be a wireless device. The data structure or queue **103** of fig. 1 illustrates an example of game setup and play. The queue is filled from top to bottom as time progresses. A sound byte for player S1 **121** is made first. Some modest pleasantries occur by player S2 **122**, S1 **123** and S4 **124**, none accompanied by application data. Other SNACs may be stored during this prelude stage **120**, such as joining the

conversation, or designating a player to go first, which need not have a sound byte component.

A game play phase **130** proceeds after prelude **120**. Player, S3, issues a SNAC bearing application data **133**. The application data is also routed to application specific logic **109** and processed. Application specific logic **109** may operate as a kind of data filter - keeping some cards hidden from some players - by adjusting the application data according to the rules of a conversation, such as, in this case, a game. Next, player S4 makes a move **134**, which includes application data. Similar processing occurs. Player S3 follows with a simple comment at SNAC **135**. Such a comment may be selected by sound byte selection **115**, and processed by multi-modal synchronization **113** to be dispatched as a SNAC without application data. Player S1 moves **151**. Player S2 moves **152**. Player S2, perhaps after a moment of thought, makes a follow-up SNAC, **155**. Player S4 comments **156**.

The embodiment may interact with a digital bearer network **160**, such as one provided by a cellular carrier, e.g. Global System for Mobiles (GSM), Cellular Packet Data Protocol (CPDP) provider, an internet service provider or a combination thereof. In any event, the digital bearer network **160** may support a non-real-time traffic class, wherein preference is given to packets marked for real-time routing. An example of a non-real-time traffic class, is the traffic supported by Short Text Messaging (SMS) on a GSM network, as compared to traffic carried on a channel of a full duplex voice communication on the GSM network. In such a case, if a radio transceiver part of the GSM network is used at full capacity for full duplex voice, any SMS traffic routed through the transceiver is queued. In other words, the transport layer that carries a SNAC may prioritize the SNAC traffic as low priority traffic, which may be transmitted only when no full duplex voice is contending for the same radio resource, e.g. a frequency.

A non-real-time delivery of a SNAC is suitable. Tolerance of delay between interactions is a function of the application environment, however, it is known that data services of a digital cellular telephone network compete with the real-time voice services, and so delays may be sufficiently low during off-peak real-time voice use. Fortuitously, the interest by users in entertainment use rises during the after-hours period from 6PM to 12PM on most cellular networks, which avoids the time of peak usage of full duplex voice-traffic during business hours.

Fig. 2 shows an example of how a SNAC record might appear, either at the server, or on the client device. A sequence number **201** may provide an indication of the order the SNAC arrived in a conversation or game. The sequence number may be used as an index into the SNAC queue at the server. A request from a client may

specify the sequence number for purposes of requesting SNACs in a non-linear order. Alternatively, a request from a client may indicate a relative sequence number, i.e. indicating that the requested SNAC is one more than the one previously sent to the client. Command **203** may be a command of a client, such as to only send a SNAC; send and request next SNAC; or a request to filter SNACs sent to the client. Commands are discussed in detail in Fig. 3. Timestamp **205** may be a time that the SNAC was sent. Timestamp **205** may include the time that a SNAC is received at either a client device or a server device. The timestamp **205** may serve as a guide to determine which SNAC is older and which SNAC is younger in the queue. ClientID **207** may be a unique identifier of the client that is the source for the SNAC. It can operate as a key upon which a process filters SNACs. GroupID **209** is a unique identifier for the instance of the conversation or game that the SNAC is relates to. Note, that for a given application, e.g. poker, there may be multiple games in progress on a server. AppID **211** is an identifier of which application the SNAC relates to. An alternative embodiment of the SNAC could combine groupID and appID into a single field.

Applen 2**13** may be an indicator of the size of the following field, App Data **215**. App data **215** may be a non-voice semantic command that is specific to the application. If the application were one of playing chess, the app data might be "pawn, to queen's rook 4", or a binary representation of such a move. SBLen **217** may be an indicator of the size of the following field, Sound Byte Data 2**19**. Sound byte data 2**19** may be voice, or other audio input that is compressed according to a codec, which may be a standard codec. A SNAC having no associated sound data, would have a SBLen **217** of zero.
A SNAC that includes voice or other audio data, is called a sound byte (SB). A SNAC that lacks voice or audio data, is called a data only SNAC (DOS).

Fig. 3 shows a communication diagram of the creation of a conversation or game. The players operate through four clients: client 1 **301**, client 2 **302**, client 3 **303**, and client 4 **304**. The server **309** operates to modify SNACs, in some cases to reflect game progress by providing new information as app data, and in other cases to also insert selected sound byte data. Optionally, the server **309** may filter SNACs according to previous instructions of a client, wherein those instructions may be to mute a game-player that is disagreeable to the client, or to apply distortion or other sound effects on a sound byte.

A conversation may be a game. Game players may logon or otherwise select the electronic address of a server, which may have a WAP interface. To prepare for the start of a game, client 2 **302** issues a command in a SNAC which includes the

command to "create or join", which may be represented by a bit pattern, or a toggled bit in a position of significance. The SNAC may be created with attendant voice. The choice of whether to send voice may be controlled using a user interface at the client. Server **309** receives the "create or join" command **311** and sends an acknowledgement **313** back to client 2 **302**. The acknowledgement **313** may include an indication of whether the client has any special privileges, i.e. to moderate communications of the group. In addition, the acknowledgement **313** may include current status information of the group. Coordination of the start of a conversation or game may take place by more conventional means, such as over the telephone, by email, or through instant messaging. Such applications may be integrated into a client device with an embodiment of the invention.

Continuing set up is accomplished when Client 2 **302** sends a SNAC including a command to "modify group parameters" **315**, which includes a timeout value setting. The timeout may operate as the time duration that all commands sent to the group will remain in force. I.e. a command may operate to prospectively operate on any next SNAC to be delivered to the server **309** relating to the conversation. If a timer indicates that the timeout period has expired after the timestamp of the SNAC bearing the command, then the command is no longer in force.

Fig. 4 shows an example of progress of a conversation or game. The client 2 **402** records voice or other audio data into a SNAC. Upon the user at client 2 **402** actuating input through a user interface, which may include a button, the client 2 performs the SEND **411** operation, of the SNAC, which may include transmitting the SNAC over a wireless carrier to a server **409**. Client 1 performs similar steps to make a second SNAC, which is sent to server **409** using the SEND **413** operation. Client 1 is the source client device for the second SNAC. Client 3 **403** makes a request **415**, which may be a packet dispatched wirelessly from the client upon actuation of an input of a user interface. Client 3 is the requesting client device for the request **415**, and any SNAC provided by the server **409** in response thereto. As with all conversation communications, the request **415** is directed to the server **409**. The server **409** has a pointer or other reference to a SNAC associated with client 3 **403**. Since, in this case, the client has not requested any SNACs until now, the pointer refers to the SNAC that is oldest in time in the data structure of the server. Server **409** looks up the SNAC, and does a SEND **417** of the SNAC1. The server **409** may advance the pointer associated with client 3 along the queue to the next SNAC in the data structure, if one exists.

Client 4 **404** makes a REQUEST **419** in a similar manner. At first the pointer

associated with client 4 points to SNAC1, so the server **409** performs a SEND **421** operation on the SNAC1. Notice that this is the same SNAC that was sent by the SEND **417** that was responsive to client 3 **403**. A subsequent REQUEST **423** from client 4 **404** generates a SEND **425** of SNAC2 in return, reflecting the new position of the pointer associated with client 4 **404**.

Client 1 **401** makes a send with request, SENDwR **427**. The SENDwR has dual functionality: it commands the client to send a SNAC over a wireless carrier, and it commands the server **409** to reply with a SNAC as soon as a SNAC is available using the pointer associated with the client. Since client 1 **401** has obtained no SNACs yet, server **409** has a pointer to SNAC1 associated with client 1 **401**. Server **409** performs the SEND operation **429** to send SNAC1 to client 1 **401**. Client 1 **401** is the source client device for the SNAC carries the SENDwR **427** command. Client 1 **401** is the requesting client device for any SNAC referred to at the server by a pointer associated with client 1 **401**.

Client 4 **404** continues with a SENDwR **431**, thereby delivering SNAC4 for storage at the server **409** in the server SNAC data structure. Server **409** replies by sending the SNAC that the pointer associated with client 4 points to, SNAC3. The server advances the pointer associated with client 4, but not to SNAC4, since the client 4 **404** is well aware of its contents. The server may have a sound byte selection pointer point to null, to indicate that the server **409** is awaiting a fresh SNAC, not sourced from client 4 **404**. A subsequent REQUEST **435** by client 4 **404** causes a countdown of time to be initiated with respect to client 4 at the server **409**. If a SNAC from any other client appears in the data structure of server **409**, before expiration of the timer, then the REQUEST **435** is fulfilled by the server **409** performing SEND on the SNAC. Fig. 4 illustrates the occurrence of the timer expiring, according to the timeout value previously set. A negative acknowledgement (NAK) **437** may be transmitted from server **409** to client 4 **404** to indicate this has occurred. By distinguishing SNACs sourced from a client, and sending only SNACs originating from other clients in the conversation, server **409** delivers SNACs to vicarious clients, i.e. clients that, in relation to the SNAC, are not the source of the SNAC. In this respect, the server **409** filters out SNACs from the SNAC queue so that with respect to sending SNACs to a first client, only SNACs originating from other clients are transmitted to the first client.

Following expiration at the server, of the timer associated with client 4 **404**, client 2 **402** makes a REQUEST **439**. The server initiated a pointer associated with client 2 when SNAC2 was received. A pointer associated with a client, may not refer to a SNAC that was received from the client. This leads to greater efficiency in

utilization of the common media of the wireless spectrum, or the common media of a wired WAN or other wired network bandwidth. Server **409** does a SEND operation **441** to transmit SNAC2 to client 2 **403.**

Fig. 5 shows the operation of a client according to an embodiment of the invention. The client **500** may be a wireless device having a CPU and local memory. The client may have a user input device **501** through which sound bytes may be recorded. The input device **501** may include a microphone. Alternatively, a keypad may also be provided as part of the input device for input of application data. Speaker, or other sound transducer **503**, provides sound output, on, for example, received SNACs. Speaker **503** may also provide a playback facility for stored voice data. Visual rendering **507** may be a display capable of providing visual indications. At its most basic level, visual rendering may be LEDs that indicate the application data. Alternatively, LCDs capable of handling graphic images and icons may be used as visual rendering **507**.

Receipt of a SNAC may be received by a radio frequency transceiver **518** also known as a wireless interface. The transceiver **518** may receive and transmit packets encoded on a wireless carrier to a WAP gateway **519.** The transceiver provides baseband data to a WAP stack **517** within the client. WAP stack **517** provides transport layer support to the filtering and application functions, which may include packet assembly and disassembly. SNAC Protocol **515** may be a program that operates as a state machine to format SNAC packets based on user-made commands. SNAC Protocol **515** may also handle the queuing of SNACs, which may include embedded commands, to the WAP stack and from the WAP stack. Filtering **513** is programmable by the user of the client device **500** to reduce some of the content of incoming SNACs. A user may desire that sound bytes are not played, or otherwise truncated, and thus program the filter **513** accordingly. This might be helpful if another player has started to transmit objectionable voice or noise. Alternatively the filter **513** may be programmable to exclude playback of audio portions of SNACs that relate to questions, as may occur, e.g. in an application supporting a lecture given by a teacher.

Application **509** is a program selected by the user of the client **500** to entertain or instruct. Application **509** must be compatible to the ALS **109** of Fig. 1, i.e., application **509** must have a common set of rules for display, game-play, scoring, to name a few. Application runs on a CPU local to client **500**, and may provide for localized feedback to input by a user by controlling the visual rendering **507.** Application **509**, may be set to quickly pass the audio portion of sound bytes to the sound rendering **505**, or to queue a sound byte in the event that one is already being

played by the sound rendering **505**. Sound rendering is paused when a sound byte is being recorded for transmittal from the client. The application data may be similarly paused from being presented until the sound data associated therewith is presented.

It is possible that a user may be in the process of recording a sound byte, in which case the application **509** may store the sound byte as a SNAC. Half duplex may be accomplished by recording a sound byte of the user, at the same time a SNAC is received from the WAP gateway **519**. If a user is not recording a sound byte through the user input device **501**, application **509** may play the sound byte received using the sound transducer **503**. The ability to store a SNAC inbound from the WAP gateway **519** is a significant aspect of the invention, since it reduces the possibility of confusion when a user is recording a sound byte for an outbound SNAC. The Wireless Application Environment (WAE) **511** provides the language application environment for execution. The application may be steps of a program written for interpretation by WAE into machine instructions native to the CPU of the client. The program may be written in Java or wireless mark-up language (WML).

Fig. 6 shows the state machine implementation of the SNAC protocol on the client. The initial state is IDLE **601**. When a user issues a command **603**, by using the user interface, he triggers a transition to the command interpret state **605**. In the example of Fig. 5, an initial command of "Create or Join" is a send **607** command, which places the client into a SENDING state **609**. Completion of the send returns the client to a loop between COMMAND INTERPRET **605** and IDLE **601**. A command that is a "send with request" (SENDwReq) causes transition to the REQUEST **615**. Transitions from REQUEST **615** may be a response from the server **619** or a timeout **621** of any previously set or default timeout period. When the server responds, the client shifts to a RECEIVING state **625**. Upon completion **627**, client may enter a loop between COMMAND INTERPRET **605** and IDLE **601**. The occurrence of a request **629** command transitions the client from COMMAND INTERPRET **605** to REQUEST **615**.

The application process, first described in Fig. 5, may await a signal from the client protocol state machine that occurs when a complete SNAC is received **627**. At that moment, the application may retrieve from storage, the audio portion of a received SB and the data portion of the SB, and provide the data portion to the visual rendering application and the audio portion to the sound rendering application nearly simultaneously. In so doing, the client synergistically depicts progress in the conversation or game through nearly simultaneous changes in a display and sound output.

Although the invention has been described in the context of particular embodiments, various alternative embodiments are possible. The client embodiments may operate within a number of different packages, e.g., a mobile phone, pager, or electronic organizer. The server may support client programmable filters, wherein the activity of selecting out or censoring of SNACs occurs at the server side, rather than the client. Such filtering naturally would be responsive to the commands of a client. The underlying wireless channel that may carry a SNAC as a bearer service may be one based on frequency division multiple access; time division multiple access or code division multiple access. A stream of packets that carry a SNAC may first be carried by a first wireless channel, and then changed to second wireless channel during the wireless transmission of the stream. Thus, while the invention has been particularly shown and described with respect to specific embodiments thereof, it will be understood by those skilled in the art that changes in form and configuration may be made therein without departing from the scope and spirit of the invention.

What is claimed is:

1. A method for mediating delivery of a Serialized Network Asynchronous Communication (SNAC) to at least one requesting client device that has made a request comprising the steps of:

storing the SNAC in storage;

filtering the SNAC according to an application to produce a second SNAC having application data and audio data; and

transmitting the second SNAC via a packet switched network to the at least one requesting client device.

2. The method of claim 1 wherein the step of filtering comprises the step of removing audio data to produce a second SNAC.

3. The method of claim 1 wherein the step of storing is preceded by the step of receiving the SNAC.

4. The method of claim 3 wherein the step of receiving comprises assembling a SNAC from at least two packets.

5. The method of claim 1 wherein storing comprises the step of storing the SNAC in a position relative to a sound byte selection pointer.

6. The method of claim 1 wherein the step of transmitting comprises the step of addressing the SNAC to all requesting client devices except a source client device.

7. The method of claim 6 wherein the step of transmitting comprises the steps of:

transmitting the SNAC to a wireless transceiver; and

transmitting the SNAC over a wireless carrier.

8. The method of claim 6 wherein the step of transmitting the SNAC over a wireless carrier comprises the step of queuing the SNAC behind full duplex voice traffic.

9. A method for presenting audio data of a SNAC and application data of a SNAC comprising the steps of:

receiving on a wireless interface a SNAC transported by at least two packets;

rendering the audio data; and

rendering the application data in synchrony with the rendering of the

audio data.

10. The method of claim 9 wherein the step of receiving further comprises the step of storing the SNAC in storage.

11. The method of claim 10 wherein the step of receiving is preceded by the step of transmitting a request to a server.

12. The method of claim 11 wherein the step of transmitting a request to a server further comprises the step of transmitting a SNAC to the server.

13. The method of claim 8 wherein the SNAC comprises data identifying a relative position of a requested SNAC.

14. A client for presenting audio data of a SNAC and application data of a SNAC comprising:

a transceiver means for wirelessly receiving a SNAC transported by at least two packets;

a sound rendering means operatively coupled to the transceiver means, said sound rendering means making a sound; and

a application rendering means operatively coupled to the transceiver means, said sound rendering means providing application data in synchrony with the sound.

15. The client of claim 14 wherein the transceiver means further comprises a storage means for storing the SNAC.

16. The client of claim 14 further comprising:

a transmitter means for transmitting a request to a server.

17. The client means of claim 15 wherein the request is at least one bit in a field of a command.

18. A server for mediating delivery of a Serialized Network Asynchronous Communication (SNAC) to at least one requesting client device that has made a request comprising:

a storage means for storing the SNAC;

a filter means for filtering the SNAC according to application to produce a second SNAC having application data and audio data; and

a transceiver means for transmitting the second SNAC via a packet switched network to the at least one requesting client device.

19. The server of claim 18 wherein the filter means comprises a means of removing audio data to produce a second SNAC.

20. The server of claim 19 further comprising:

a receiver means for receiving the SNAC operatively coupled to the storage means.

21. The server of claim 20 wherein the receiving means further comprises:

an assembly means for assembling a SNAC from at least two packets.

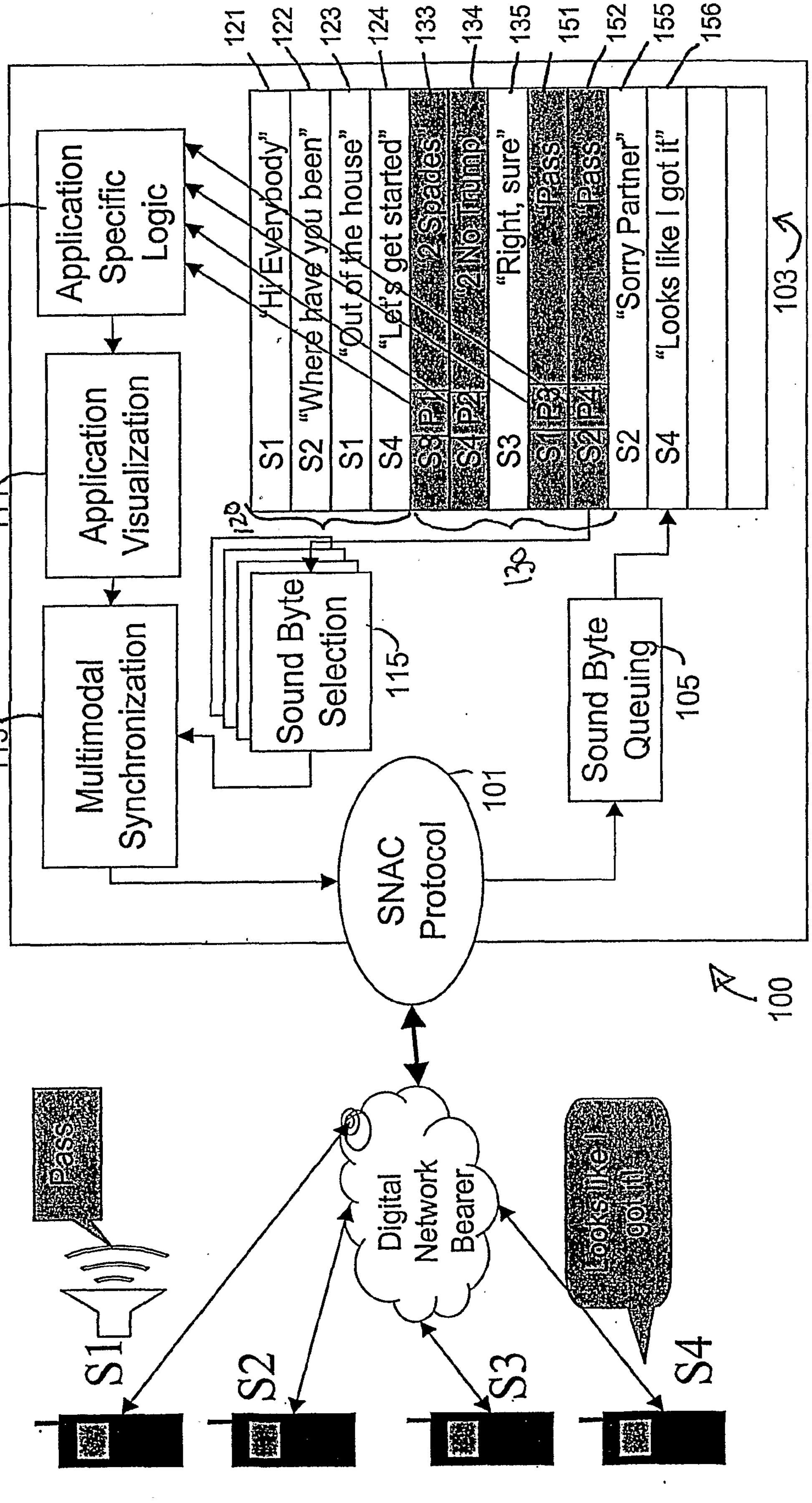
22. The server of claim 21 wherein the transceiver means further comprise:

an addressing means for addressing the SNAC to all requesting devices except a source client device.

23. The server of claim 22 wherein the transceiver means further comprises:

a wired transmitter means for transmitting the SNAC to a wireless transmitter; and

a wireless transmitter means for transmitting the SNAC over a wireless carrier.



Application Specific Logic
109
Application Visualization
111
Multimodal Synchronization
113
Sound Byte Selection
115
Sound Byte Queuing
105
SNAC Protocol
101
Digital Network Bearer
S1
S2
S3
S4
Pass
Looks like I got it
100
103
120
130
S1 "Hi Everybody"
S2 "Where have you been"
S1 "Out of the house"
S4 "Let's get started"
S3 P1 2 Spades
S4 P2 2 No Trump
S3 "Right, sure"
S1 P3 Pass
S2 P4 Pass
S2 "Sorry Partner"
S4 "Looks like I got it"
121
122
123
124
133
134
135
151
152
155
156

Fig. 1

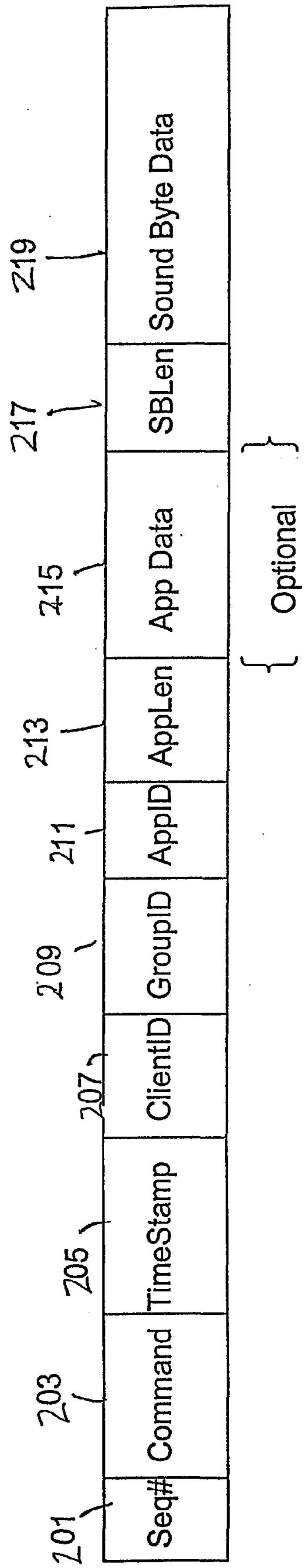

201
Seq#
203
Command
205
TimeStamp
207
ClientID
209
GroupID
211
AppID
213
AppLen
215
App Data
217
SBLen
219
Sound Byte Data
Optional

Fig. 2

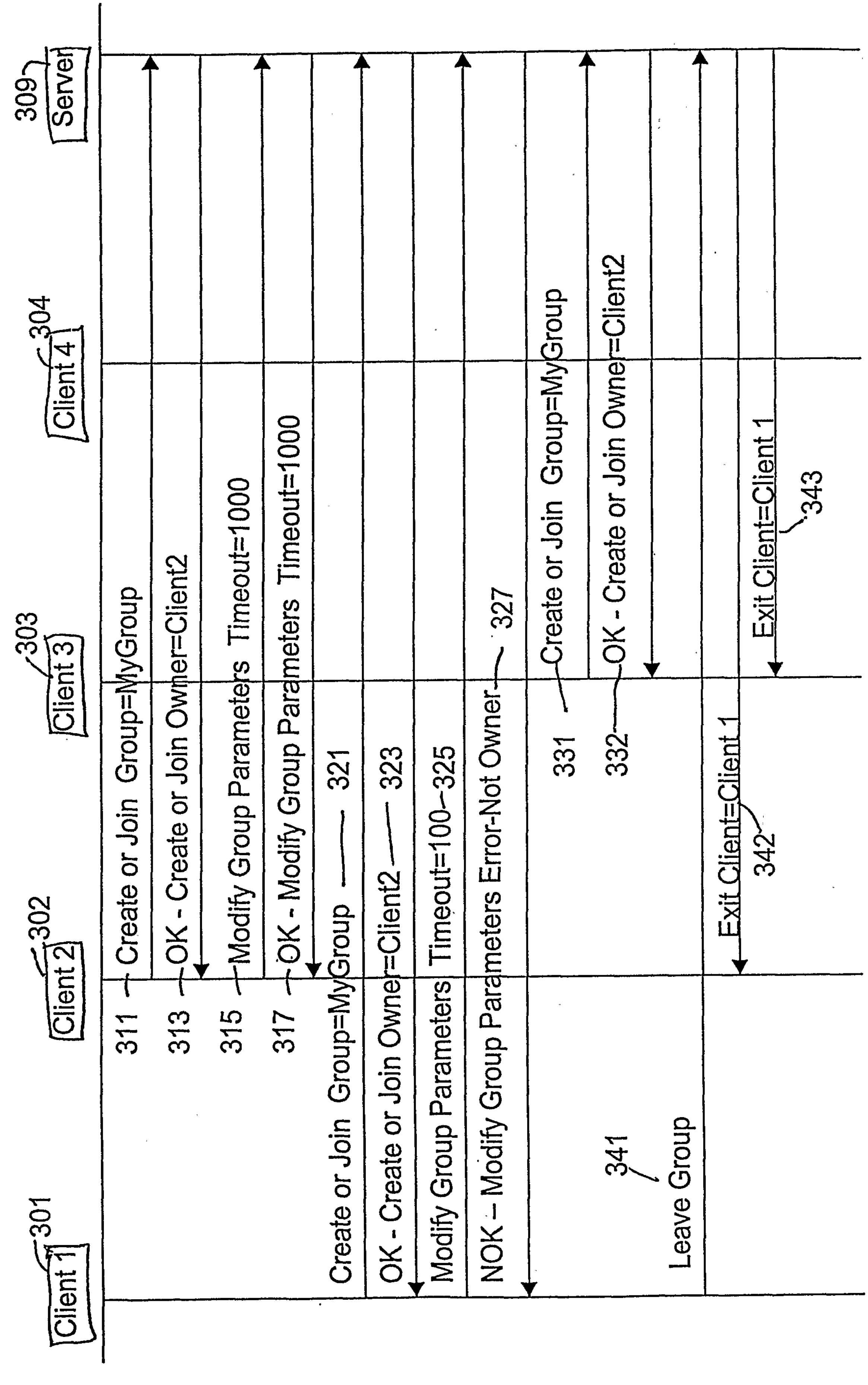

Client 1
301
Client 2
302
Client 3
303
Client 4
304
Server
309
311 Create or Join Group=MyGroup
313 OK - Create or Join Owner=Client2
315 Modify Group Parameters Timeout=1000
317 OK - Modify Group Parameters Timeout=1000
Create or Join Group=MyGroup 321
OK - Create or Join Owner=Client2 323
Modify Group Parameters Timeout=100 325
NOK – Modify Group Parameters Error-Not Owner 327
Create or Join Group=MyGroup 331
OK - Create or Join Owner=Client2 332
341 Leave Group
Exit Client=Client 1 342
Exit Client=Client 1 343

Fig. 3

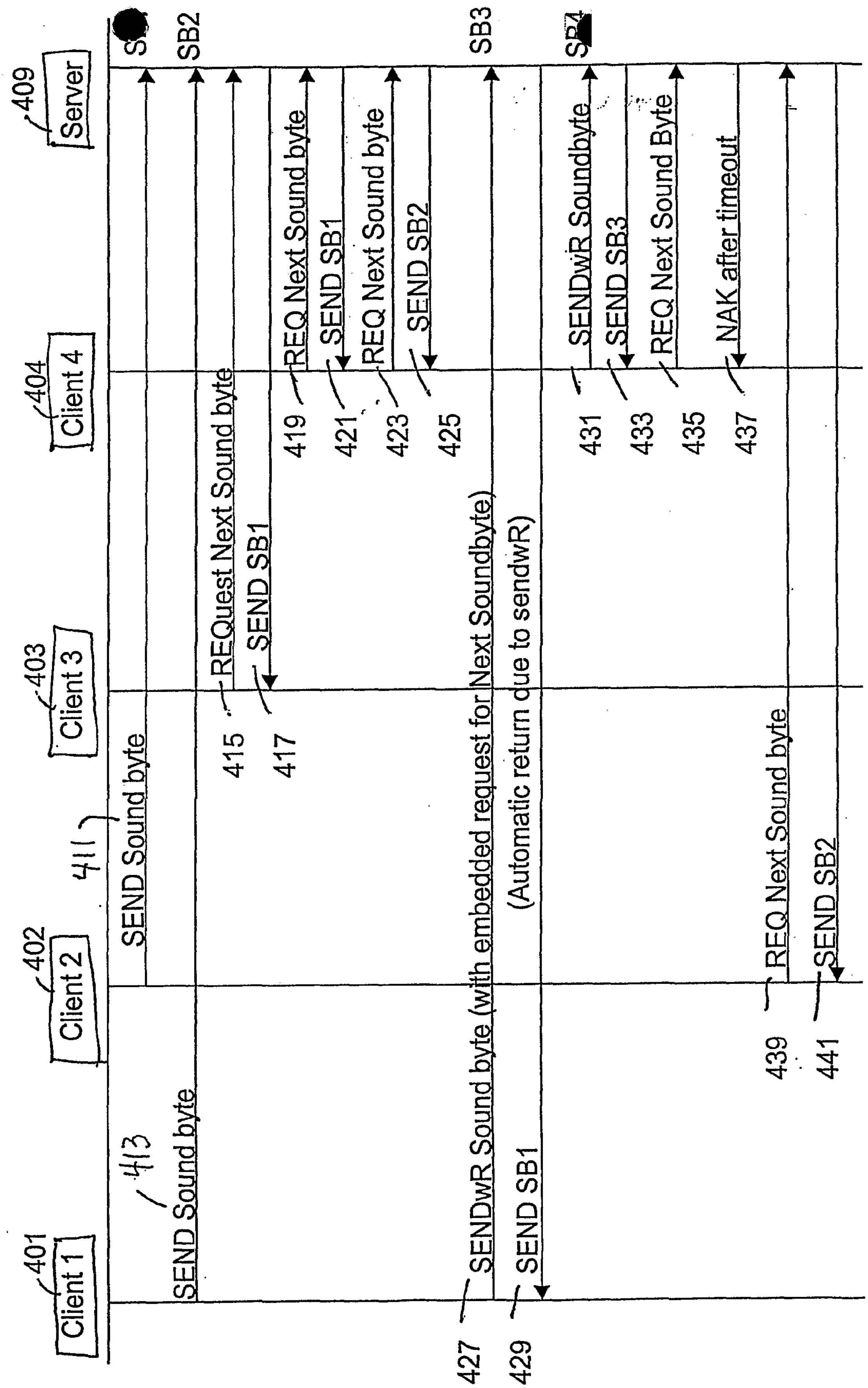

Client 1
401
Client 2
402
Client 3
403
Client 4
404
Server
409
SB2
SB3
SEND Sound byte
411
SEND Sound byte
413
REQuest Next Sound byte
415
SEND SB1
417
REQ Next Sound byte
419
SEND SB1
421
REQ Next Sound byte
423
SEND SB2
425
SENDwR Sound byte (with embedded request for Next Soundbyte)
427
SEND SB1
429
(Automatic return due to sendwR)
SENDwR Soundbyte
431
SEND SB3
433
REQ Next Sound Byte
435
NAK after timeout
437
REQ Next Sound byte
439
SEND SB2
441

Fig. 4

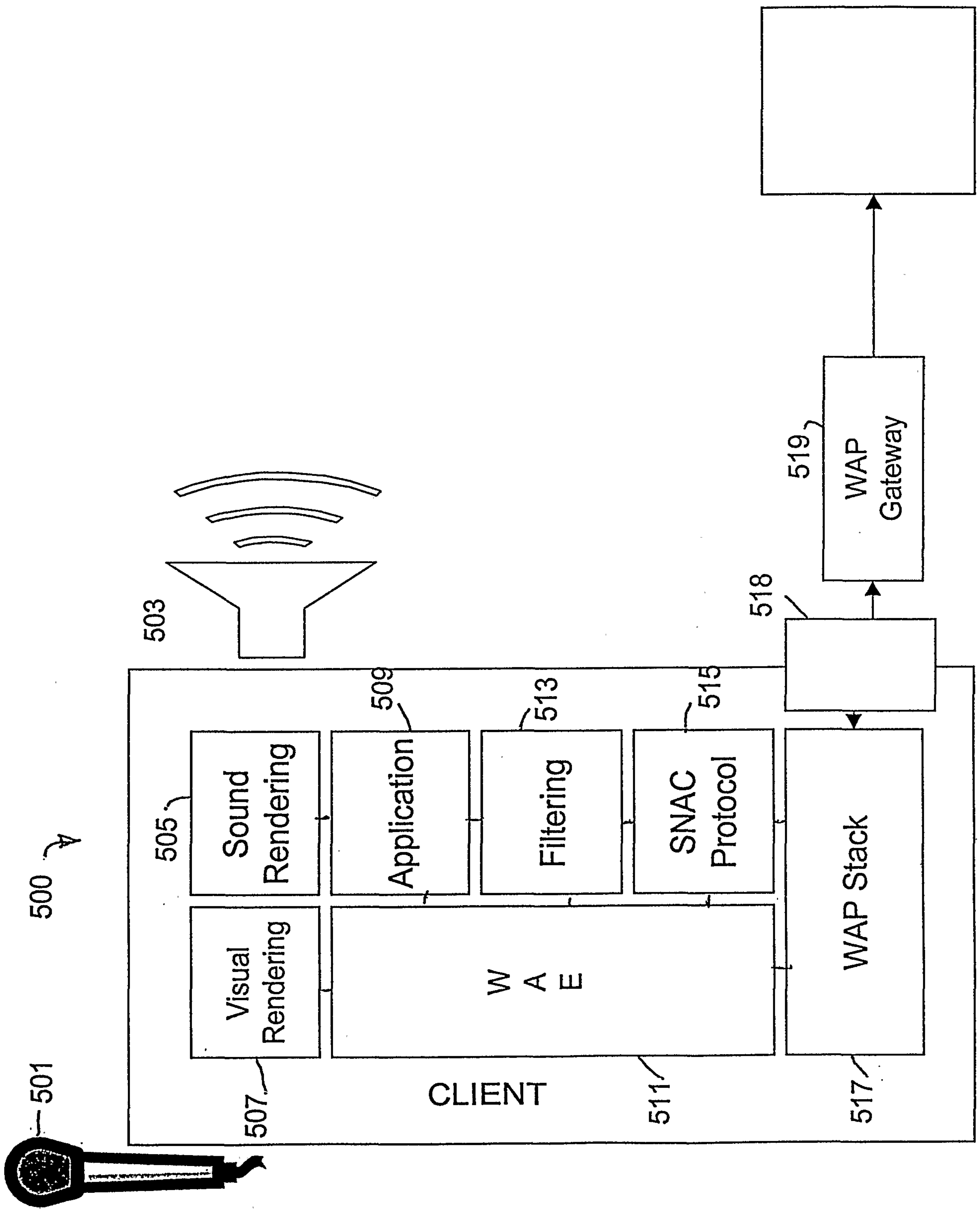

500
501
503
505
Sound Rendering
507
Visual Rendering
509
Application
511
W A E
513
Filtering
515
SNAC Protocol
517
WAP Stack
518
519
WAP Gateway
CLIENT

Fig. 5

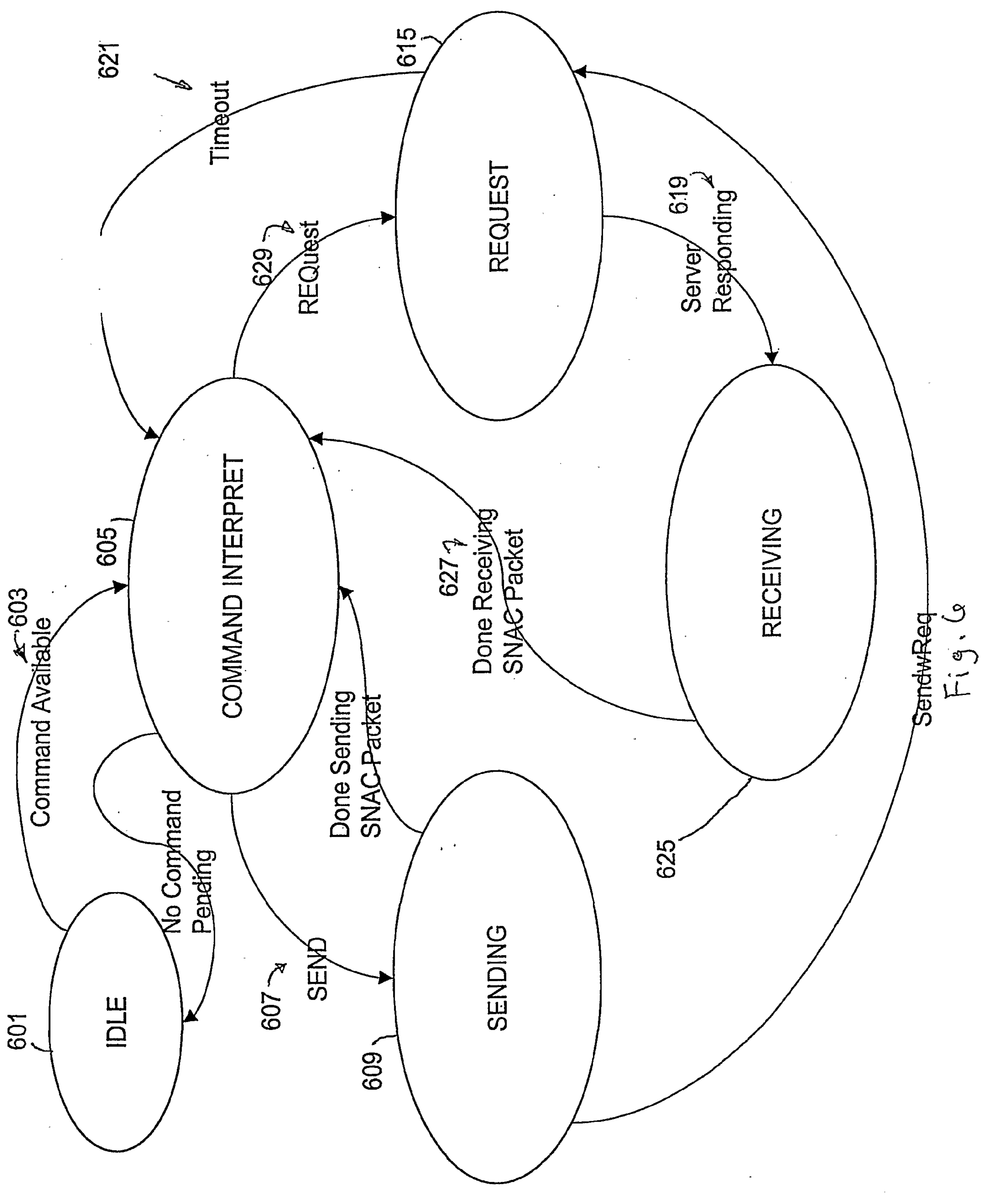

IDLE
601
COMMAND INTERPRET
605
REQUEST
615
RECEIVING
625
SENDING
609
Command Available
603
No Command Pending
SEND
607
REQuest
629
Timeout
621
Server Responding
619
Done Receiving SNAC Packet
627
Done Sending SNAC Packet
SendwReq

Fig. 6